Statistical Techniques in Robotics (16-831, S21) Lecture #14 (Wednesday, March 24)

EXP3, EXP4

Lecturer: Kris Kitani *Scribes: Yen-Chi Cheng, Hao-Ming Fu*

1 Review

In the last lecture, we talked about the Thompson Sampling [2] in the stochastic multi-armed bandits problem. In this setting, we assume that each reward is drawn from a parametrized distribution $r \sim p(r|a,\theta)$. If the true parameters for the likelihood function are known, our best action to pull the arm is

$$a = \arg\max_k \mathbb{E}_{p(r|a_k,\theta_k^*)[r|a_k,\theta_k^*]}$$

but often the true parameters are unknown, so we approximate it with the posterior distribution

$$\hat{\theta} = \arg\max_\theta p(\theta|h^{(t)} = (a^{(1)}, r^{(1)}, \dots, a^{(t)}, r^{(t)})) \tag{1}$$

where $h^{(t)}$ is the history. To model the posterior distribution, we apply the Bayes' Rule:

$$\begin{aligned} p(\theta|a^{(1)}, r^{(1)}, \dots, a^{(t)}, r^{(t)}) &= \frac{\prod_t p(r^{(t)}|a^{(t)}, \theta)p(\theta)}{\prod_t p(r^{(t)})|a^{(t)})} \\ &\propto \prod_t p(r^{(t)}|a^{(t)}, \theta)p(\theta) && \text{by Markov Assumption} \\ &\propto p(r^{(t)}|a^{(t)}, \theta)p(\theta|h^{(t-1)}) && \text{writing the posterior recursively} \end{aligned}$$

Now back to the original problem, the Equation 1 becomes:

$$\hat{\theta}_k = \arg\max_{\theta_k} p(r^{(t)}|a_k^{(t)}, \theta_k)p(\theta_k|h_k^{(t-1)})$$

To update the parameter efficiently, we adopt the conjugate priors: given that the Beta distribution is the conjugate prior of the Bernoulli, the update equation w.r.t a single arm is

$$\begin{aligned} p(\theta|h^{(t)}) &\propto p(r^{(t)}|a_k^{(t)}, \theta_k)p(\theta_k|h_k^{(t-1)}) \\ &\propto \theta^r(1-\theta)^{(1-r)}\theta^{(\alpha-1)}(1-\theta)^{(\beta-1)} \\ &\propto \theta^{(r+\alpha)-1}(1-\theta)^{(1-r+\beta)-1} \\ &\propto \theta^{\alpha'-1}(1-\theta)^{\beta'-1} \end{aligned}$$

The algorithm is listed at Alg. 1. And the regret of the Thompson Sampling is
$BR(T) = O(\sqrt{KT\log T})$.

Algorithm 1 Thompson Sampling.

```
function THOMPSONS SAMPLING
    for $t = 1, \cdots, T$ do
        $\theta_k \sim p(\theta_k|h_k^{(t')}) \quad \forall k$                                   ▷ sample from posterior
        $a_{\hat{k}}^{(t)} = \arg\max_k \mathbb{E}_{p(r|a_k,\theta_k)}[r|a_k, \theta_k]$          ▷ predict
        RECEIVE($r^{(t)}$)                                                  ▷ get sampled reward
        $h_{\hat{k}}^{(t'+1)} = h_{\hat{k}}^{t'} \bigcup (r^{(t)}, a_{\hat{k}}^{(t)})$            ▷ update history
        $p(\theta_{\hat{k}}|h_{\hat{k}}^{(t'+1)}) \propto p(h_{\hat{k}}^{(t'+1)}|\theta_{\hat{k}})p(\theta_{\hat{k}}|h_{\hat{k}}^{(t'+1)})$   ▷ update posterior
    end for
end function
```

2 EXP3

2.1 Overview

EXP3 [1] is a multi-armed bandit problem in an adversarial environment setting. The environment could observe the agents' actions and decides the payoff structure based on it. The adversarial setting is one of the strongest generalizations of the bandit problem. Unlike the stochastic environment, it removes all the assumptions of the reward followings a certain distribution and thus provides a more generalized solution. The EXP3 stands for **Exp**onential-Weight Update algorithm for **Exp**loration and **Exp**loitation. Different types of bandits are listed at Table 1.

The algorithm is listed at Algorithm 2. One can notice that EXP3 algorithm is very similar to the Hedge algorithm. The differences are as follows:

- EXP3 do not have the *expert advice*.
- We receive *partially observable reward*, while in the Hedge algorithm, the true label is revealed.
- The loss is known for *one* expert, and could only update *one* weight, while the hedge algorithm could update all weights since the loss is known for all hypothesis.

	Context-free	**Contextual**
Stochastic environment	Explore-Exploit, UCB, Thompson Sampling	linUCB
Adversarial environment	**EXP3**	**EXP4**

Table 1: Types of bandits

Algorithm 2 EXP3 Algorithm

```
1:  function EXP3($\gamma \in [0,1]$)
2:      $\boldsymbol{w}^{(1)} \leftarrow \{w_k^{(1)}\}_{k=1}^{K}$                          ▷ weights over actions
3:      for $t = 1, \cdots, T$ do
4:          $\boldsymbol{p}^{(t)} = \frac{\boldsymbol{w}^{(t)}}{\sum_k w_k^{(t)}}$               ▷ probability over actions
5:          $k \sim$ MULTINOMIAL($\boldsymbol{p}^{(t)}$)                  ▷ take and draw action
6:          $a^{(t)} = a_k$
7:          RECEIVE ($r^{(t)} \in [0,1]$)                            ▷ get reward
8:          $w_k^{(t+1)} \leftarrow w_k^{(t)} \exp(\gamma \cdot r^{(t)}/p_k^{(t)})$         ▷ update weights
9:      end for
10: end function
```

2.2 The Unbiased Estimator

In line 8 of the Algorithm 2, we notice an mysterious update term $r^{(t)}/p_k^{(t)}$. The re-weighting is the sampling strategy to make the sampled reward an *unbiased estimator*. To understand this, we discuss two cases below:

- **Case 1: Single-arm stochastic bandit.** To estimate the true mean μ_t, we sampled the reward and compute the empirical mean reward by defining the following estimator:

$$c_1^{(t)}(a^{(t)}) = \mathbf{1}[a^{(t)} = 1] \cdot r^{(t)} \,.$$

  where $c_1^{(t)}$ denotes the first arm at time t. And the expected value of $c_1^{(t)}(a^{(t)})$ at time t is

$$\begin{aligned} \mathbb{E}_{p(a)}\left[c_1^{(t)}(a^{(t)})\right] &= p(a^{(t)} = 1) \cdot \mathbf{1}[a^{(t)} = 1] \cdot r^{(t)} \,, \\ &= r^{(t)} \,. \end{aligned}$$

  The expected value of the estimator over T round:

$$\mathbb{E}_{p(a)}\left[\frac{1}{T}\sum_{t=1}^{T} c_1^{(t)}(a^{(t)})\right] = \frac{1}{T}\sum_{t=1}^{T} r^{(t)} \,.$$

  which is the empirical estimate of the true mean μ_t.

- **Case 2: Two-arm stochastic bandit**. In this case, we could select action from two arms with a probability distribution (50/50). We again compute the empirical mean for arm 1 as follows:

$$\begin{aligned} c_1^{(t)}(a^{(t)}) &= \mathbf{1}[a^{(t)} = 1] \cdot r^{(t)} \,, \\ c_2^{(t)}(a^{(t)}) &= \mathbf{1}[a^{(t)} = 2] \cdot r^{(t)} \,. \end{aligned}$$

  where c_k denotes k_{th} arm, $k = 1, 2$. And the expected value of the estimator is

$$\begin{aligned} \mathbb{E}_{p(a)}\left[c_1^{(t)}(a^{(t)})\right] &= p(a^{(t)} = 1) \cdot c_1^{(t)}(1) + p(a^{(t)} = 2) \cdot c_1^{(t)}(2) \,, \\ &= p(a^{(t)} = 1) \cdot \mathbf{1}[a^{(t)} = 1] \cdot r^{(t)} + p(a^{(t)} = 2) \cdot \mathbf{1}[a^{(t)} = 1] \cdot r^{(t)} \,, \\ &= (0.5) \cdot 1 \cdot r^{(t)} + (0.5) \cdot 0 \cdot r^{(t)} \,, \\ &= (0.5) \cdot r^{(t)} \,. \end{aligned} \tag{2}$$

  here we are getting a scaled reward back. We continue to compute the expected value of the estimator over T round:

$$\mathbb{E}_{p(a)}\left[\frac{1}{T}\sum_{t=1}^{T} c_1^{(t)}(a^{(t)})\right] = \frac{1}{T}\sum_{t=1}^{T} (0.5) r^{(t)} \,.$$

  and we found that the reward is multiplied by the probability of that arm. To fix this, *inverse probability weighting* or *importance sampling* is proposed to address this problem.

2.3 Inverse Probability Weighting

Recall Equation 2, the expected value of the estimator is scaled down by 0.5. To fix that, we multiply the term by *the inverse of the probability of selecting that arm* (which is $\frac{1}{p_1} = 2$):

$$c_1^{(t)}(a^{(t)}) = \mathbf{1}[a^{(t)} = 1] \cdot r^{(t)} \cdot \frac{1}{p_1} \,,$$

and the expected value of the estimator becomes

$$
\begin{aligned}
\mathbb{E}_{p(a)}\left[c_1^{(t)}(a^{(t)})\right] &= p(a^{(t)}=1)\cdot c_1^{(t)}(1) + p(a^{(t)}=2)\cdot c_1^{(t)}(2)\,, \\
&= p(a^{(t)}=1)\mathbf{1}[a^{(t)}=1]\cdot r^{(t)}\cdot\frac{1}{p_1} + p(a^{(t)}=2)\mathbf{1}[a^{(t)}=1]\cdot r^{(t)}\cdot\frac{1}{p_1}\,, \\
&= (0.5)\cdot 1\cdot r^{(t)}\cdot\frac{1}{0.5} + (0.5)\cdot 0\cdot r^{(t)}\cdot\frac{1}{0.5}\,, \\
&= r^{(t)}\,.
\end{aligned}
$$

so the expected value is the reward value, and the estimator becomes the unbiased estimator to the true mean μ_t. In general, the unbiased estimator for the expected reward of k_{th} arm, given that the probability of selecting it equals p_k, is

$$
\begin{aligned}
c_k^{(t)}(a^{(t)}) &= \mathbf{1}[a^{(t)}=k]\cdot r^{(t)}\cdot\frac{1}{p_k}\,,\ \ \forall k\,, \\
\mathbb{E}\left[c_a^{(t)}(a^{(t)})\right] &= \sum_{k=1}^{K} p_k\cdot\mathbf{1}[a^{(t)}=a]\cdot r^{(t)}\cdot\frac{1}{p_k}\,, \\
&= p(a^{(t)}=a)\cdot r^{(t)}\cdot\frac{1}{p_a}\,, \\
&= r^{(t)}\,.
\end{aligned}
$$

Therefore the mysterious term in line 8 of the Algorithm 2, is the unbiased estimator to the true mean μ_t. Now let's discuss the behavior of the update equation:

$$w_k^{(t+1)} = w_k^{(t)}\exp\{\gamma\cdot r^{(t)}/p_k^{(t)})\}$$

Given the reward $r^{(t)}$ and the probability $p_k^{(t)}$, the behaviors of the agent are listed at Table 2

Reward r	**Probability** p	**Results in...**
huge	tiny	Exploitation
small	tiny	Exploration
0	moderate	unchanged

Table 2: *Exploration* or *Exploitation*?

2.4 Regret Bound

Now we try to bound the regret for the EXP3 algorithm. As usual, we first properly define a potential function that assists our mathematical calculation. Then, we finds the upper and lower bounds of the potential function. Finally, we combine the inequalities and apply some basic algebra to derive the regret bound.

First, for convenience, we define $z^{(t)}$ as the sum of the weights w of all k bandits at time t:

$$z^{(t)} = \sum_k w_k^{(t)} \tag{3}$$

With this, now we define the potential function $\Phi()$ as the sum of log difference of $z^{(t)}$ overall all time steps until time T:

$$\Phi = \sum_{t=1}^{T} log(\frac{z^{(t+1)}}{z^{(t)}}) \tag{4}$$

Then, we calculate the upper bound of this potential function Φ. By inserting the definition of z into that of Φ, we have:

$$\Phi = \sum_{t=1}^{T} log(\frac{z^{(t+1)}}{z^{(t)}}) = \sum_{t=1}^{T} log(\frac{\sum_k w_k^{(t+1)}}{\sum_{k'} w_{k'}^{(t)}}) \tag{5}$$

Assuming that we are dealing with a single known sequence, recall that from time t to $t+1$, we update each weight w by multiplying it with an exponential term that conditions on the gain g of each bandit. Thus, we can get this:

$$\Phi = \sum_{t=1}^{T} log(\frac{\sum_k w_k^{(t)} exp(\gamma \cdot g_k^{(t)})}{\sum_{k'} w_{k'}^{(t)}}) = \sum_{t=1}^{T} log(\sum_k \frac{w_k^{(t)}}{\sum_{k'} w_{k'}^{(t)}} exp(\gamma \cdot g_k^{(t)})) \tag{6}$$

Notice that the fractional term with w is actually the probability of sampling the k^{th} bandit, p_k, according to the algorithm. So, we can replace the term with p and obtain:

$$\Phi = \sum_{t=1}^{T} log(\sum_k p_k^{(t)} exp(\gamma \cdot g_k^{(t)})) \tag{7}$$

To replace the exponential term with a polynomial one, we apply the inequality $e^x \leq 1 + x + x^2$ and obtain:

$$\Phi \leq \sum_{t=1}^{T} log(\sum_k p_k^{(t)} (1 + \gamma \cdot g_k^{(t)} + (\gamma \cdot g_k^{(t)})^2)) \tag{8}$$

By distributing $\sum_k p_k^{(t)}$ into each of the terms, we obtain:

$$\Phi \leq \sum_{t=1}^{T} log(\sum_k p_k^{(t)} + \sum_k p_k^{(t)} \gamma \cdot g_k^{(t)} + \sum_k p_k^{(t)} (\gamma \cdot g_k^{(t)})^2) \tag{9}$$

Knowing that the selection probabilities of each bandit should sum to 1 and the inequality $log(x + 1) \leq x$, we have

$$\Phi \leq \sum_{t=1}^{T} log(1 + \sum_k p_k^{(t)} \gamma \cdot g_k^{(t)} + \sum_k p_k^{(t)} (\gamma \cdot g_k^{(t)})^2) \tag{10}$$

$$\leq \sum_{t=1}^{T} (\sum_k p_k^{(t)} \gamma \cdot g_k^{(t)} + \sum_k p_k^{(t)} (\gamma \cdot g_k^{(t)})^2) \tag{11}$$

This is the upper bound of the potential function. Now, we calculate its lower bound. We represent Φ as the log movement of z from its initial value K:

$$\Phi = log(\frac{z^{(T+1)}}{z^{(1)}}) = log(\frac{z^{(T+1)}}{K}) \tag{12}$$

Inserting the definition of z and considering the update of w from its initial value 1, we have:

$$\begin{aligned}
\Phi &= -logK + log\sum_{k=1}^{K} w_k^{(T+1)} && (13)\\
&= -logK + log\sum_{k=1}^{K}\prod_{t=1}^{T} exp(\gamma \cdot g_k^{(t)}) && (14)\\
&= -logK + log\sum_{k=1}^{K} exp(\gamma \cdot \sum_{t=1}^{T} g_k^{(t)}) && (15)
\end{aligned}$$

As the gain g is always positive, we can arbitrarily pick any k and obtain the inequality:

$$\begin{aligned}
\Phi &\geq -logK + log(exp(\gamma \cdot \sum_{t=1}^{T} g_k^{(t)})) \quad \forall k && (16)\\
&= -logK + \gamma \cdot \sum_{t=1}^{T} g_k^{(t)} \quad \forall k && (17)
\end{aligned}$$

This is the lower bound of the potential function. Finally, we combine the upper and the lower bounds:

$$-logK + \gamma \cdot \sum_{t=1}^{T} g_j^{(t)} \leq \sum_{t=1}^{T}(\sum_{k} p_k^{(t)}\gamma \cdot g_k^{(t)} + \sum_{k} p_k^{(t)}(\gamma \cdot g_k^{(t)})^2) \quad \forall j \tag{18}$$

Notice that we change the index k in the lower bound to k to avoid the conflict with the summation index k in the upper bound. Now, rather than let the inequality hold for all bandits j, we select the one with the maximum gain to obtain the strongest statement among these options. This help us get rid of the $\forall j$ statement. Also, with some algebraic operations, we obtain

$$\max_j \sum_{t=1}^{T} g_j^{(t)} - \sum_{t=1}^{T}\sum_{k=1}^{K} p_k^{(t)} g_j^{(t)} \leq \frac{logK}{\gamma} + \sum_{t=1}^{T}\sum_{k=1}^{K} p_k^{(t)} \cdot \gamma (g_k^{(t)})^2 \tag{19}$$

Notice that the first term on the left hand side is the total gain of the best bandit and the second term is the performance of the EXP3 algorithm. Combining these, the left hand side of the inequality is the regret of EXP3, which gives us the upper bound of the regret of EXP3.

This result is obtained from a specific sequence of actions and a certain sampled reward from the adversarial world. To make the regret bound general, we have to calculate the expected value of regret rather than that of a certain sequence. Take the expectation for the both sides:

$$\begin{aligned}
E(\textbf{regret}) = R^{(T)} &\leq E(\frac{logK}{\gamma} + \gamma \sum_{t=1}^{T} \sum_{k=1}^{K} p_k^{(t)} \cdot (g_k^{(t)})^2) \\
&\leq \frac{logK}{\gamma} + \gamma \sum_{t=1}^{T} E(\sum_{k=1}^{K} p_k^{(t)} \cdot (g_k^{(t)})^2) \\
&= \frac{logK}{\gamma} + \gamma \sum_{t=1}^{T} E(p_{k^{(t)}}^{(t)} \cdot (\frac{r^{(t)}}{p_{k^{(t)}}^{(t)}})^2) = \frac{logK}{\gamma} + \gamma \sum_{t=1}^{T} E(\frac{(r^{(t)})^2}{p_{k^{(t)}}^{(t)}}) \\
&\leq \frac{logK}{\gamma} + \gamma \sum_{t=1}^{T} E(\frac{1}{p_{k^{(t)}}^{(t)}}) = \frac{logK}{\gamma} + \gamma \sum_{t=1}^{T} K = \frac{logK}{\gamma} + \gamma TK
\end{aligned}$$

Finally, we can observe that when γ goes to infinity or negative infinity, the bound explodes. As a result, we can find the γ that leads to the smallest upper bound by requiring the derivative of the bound to be zero:

$$\frac{d}{d\gamma}(\frac{logK}{\gamma} + \gamma TK) = 0 \tag{20}$$

$$\implies -\frac{logK}{\gamma^2} + TK = 0 \tag{21}$$

$$\implies \gamma^2 = \frac{logK}{TK} \tag{22}$$

$$\implies \gamma = \sqrt{\frac{logK}{TK}} \tag{23}$$

Now we have the γ that minimizes the upper bound of regret. We then calculate the corresponding upper bound by inserting this γ into the bound:

$$R \leq \frac{logK}{\sqrt{\frac{logK}{TK}}} + \sqrt{\frac{logK}{TK}}TK \tag{24}$$

$$= 2\sqrt{TKlogK} \tag{25}$$

This means that with a big O notation,

$$R = O(\sqrt{TKlogK}) \tag{26}$$

This shows that the regret is sublinear to T, meaning that EXP3 is a no regret algorithm.

3 EXP4

The EXP4 algorithm, Exponential-Weighted Update algorithm for Exploration and Exploitation with Experts, is similar to the EXP3 algorithm in many ways. The major difference is that EXP4 learns weights for each experts providing advises for pulling the bandits rather than learning weights for each bandit directly. This is the algorithm of EXP4:

Algorithm 3 EXP4 Algorithm

```
1: function EXP4($\gamma \in [0, 1], T$)
2:     $\boldsymbol{w}^{(1)} \leftarrow 1 \in R^N$                                        ▷ weights over experts
3:     for $t = 1, \cdots, T$ do
4:         Receive ($\boldsymbol{X}^{(t)} \in R^{N \times K}$)                          ▷ advice from N experts
5:         $\boldsymbol{q}^{(t)} = \frac{\boldsymbol{w}^{(t)}}{||\boldsymbol{w}||^2} \cdot \boldsymbol{X}^{(t)} \in \Delta^K$             ▷ probability over actions
6:         $k^{(t)} \sim$Multinomial($\boldsymbol{q}^{(t)}$)                                 ▷ draw action
7:         Receive ($r^{(t)}$)                                                  ▷ get reward
8:         $\hat{\boldsymbol{r}}^{(t)} = \frac{r^{(t)}}{q_k^{(t)}} I[k = k^{(t)}] \in I^K$                    ▷ reward over all arms
9:         $\boldsymbol{g}^{(t)} = \boldsymbol{X}^{(t)} \cdot \hat{\boldsymbol{r}}^{(t)} \in R^N$                          ▷ per expert reward
10:        $w_n^{(t+1)} \leftarrow w_n^{(t)} \exp(\gamma \cdot g_n^{(t)}) \quad \forall n$                 ▷ update
11:    end for
12: end function
```

From the algorithm we can observe the major differences lie in line 5 and line 10, where the probability distribution over actions are replaced by that over experts and weights for experts rather than actions are updated.
Through similar derivation process for EXP3, we can obtain that the regret bound of EXP4 has a similar form with that of EXP3.

$$R_{EXP3} \leq \sqrt{KTlogK} \tag{27}$$

$$R_{EXP4} \leq \sqrt{KTlogN} \tag{28}$$

Note that the $logK$ is now replaced by $logN$ because the number of weights has changed from the number of bandits (actions) K to that of experts N.

4 Conclusion

In this lecture, we discuss a more generalized setting in the multi-armed bandits problem: the adversarial environment. The EXP3 algorithm is proposed to solve this problem. The algorithm is similar to the Hedge algorithm we learned before, but there is no expert advises and the loss is only known to one action, and we could only update one weight. In the update equation of the EXP3 algorithm, the Inverse Probability Weighting is incorporated to make the true mean estimator $c_k^{(t)}(a^{(t)})$ an unbiased estimator. The update equation in EXP3 will make the agent to *explore* more if the sampled reward is small, and *exploit* if the sampled reward is huge.

Then we tried to obtain an upper bound for its regret. As always, we defined a potential function for which we can find an upper bound and a lower bound. Finally, we combine the two bounds and obtain the upper bound of regret. As γ is a hyper parameter that we can design, we found the *gamma* that minimizes the regret bound. With such γ, the upper bound of regret is sublinear with regard to time steps T and thus EXP3 is a no regret algorithm.

As for EXP4, knowing that it is quite similar to EXP3, we noticed that their major difference is that each of the weights in EXP3 corresponds to an action, while that in EXP4 corresponds to an expert. Accordingly, we also derived the regret bound for EXP4.

References

[1] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002. 2

[2] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. 1